

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-282461

(43)Date of publication of application : 15.10.1999

(51)Int.Cl.

G10H 1/00  
G06F 15/16

(21)Application number : 10-081323

(71)Applicant : YAMAHA CORP

(22)Date of filing : 27.03.1998

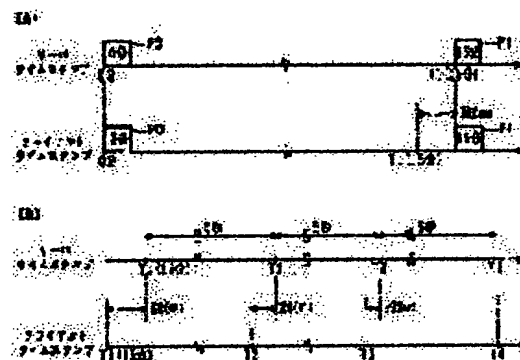
(72)Inventor : KIKUCHI TAKESHI  
KOIKE YUJI

## (54) COMMUNICATION DEVICE AND METHOD, AND MEDIUM ON WHICH PROGRAM IS RECORDED

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To increase reliability of timing information at the time of communicating timing information by arranging a time difference detection means to detect a time difference between a 1st timing information and a 2nd temporal information, and a correction means to correct the timing information generated by a timing information generation means according to a prescribed time difference or longer.

**SOLUTION:** A server transmits a packet P1 containing a time-stamped record of 15 minutes to a client after the lapse of 15 minutes. The client compares time-stamped records of both with each other at the time of receiving the packet P1, and confirms that own time-stamped record is advanced by 300 msec. Namely, the time-stamped record of the client is 300 msec. more advanced than the one of the server at a time T1. Then, the client starts correction processing of the time-stamped record at the time of recognizing that own time-stamped records have advanced, for instance, 300 msec. or longer by more than the number of a fixed data. In this case, the deviation of the time-stamped record should not be corrected at a stretch, but corrected three times by taking about 15 seconds by 100 msec.



## LEGAL STATUS

[Date of request for examination]

07.12.1999

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-282461

(43) 公開日 平成11年(1999)10月15日

(51) Int.Cl.<sup>6</sup>

G 1 0 H 1/00

G 0 6 F 15/16

識別記号

3 3 0

F I

G 1 0 H 1/00

G 0 6 F 15/16

Z

3 3 0 D

審査請求 未請求 請求項の数 4 O L (全 12 頁)

(21) 出願番号

特願平10-81323

(22) 出願日

平成10年(1998)3月27日

(71) 出願人 000004075

ヤマハ株式会社

静岡県浜松市中沢町10番1号

(72) 発明者 菊池 健

静岡県浜松市中沢町10番1号 ヤマハ株式会社内

(72) 発明者 小池 祐二

静岡県浜松市中沢町10番1号 ヤマハ株式会社内

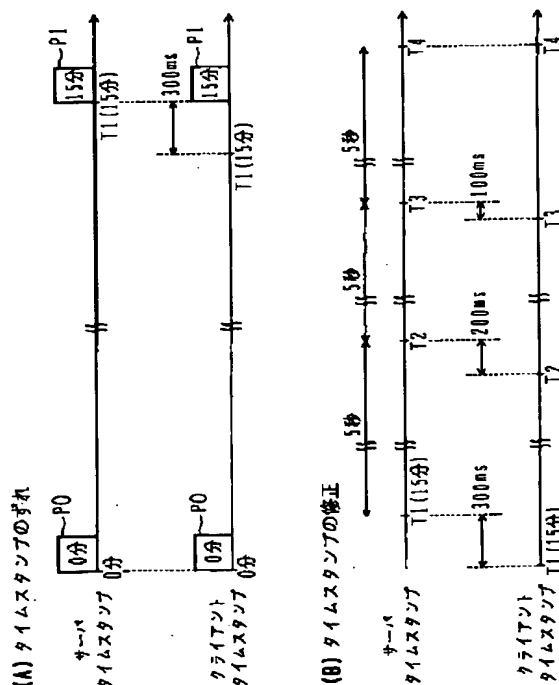
(74) 代理人 弁理士 高橋 敬四郎 (外1名)

(54) 【発明の名称】 通信装置、通信方法及びプログラムを記録した媒体

(57) 【要約】

【課題】 時間情報を通信する際に時間情報の信頼性を高めることができる通信装置、通信方法又はプログラムを記録した媒体を提供することを課題とする。

【解決手段】 第1の時間情報を外部から受信する受信手段と、第2の時間情報を生成する時間情報生成手段と、前記第1の時間情報と前記第2の時間情報との所定以上の時間差を検出する時間差検出手段と、前記所定以上の時間差に応じて前記時間情報生成手段が生成する第2の時間情報を修正する修正手段とを有する。



## 【特許請求の範囲】

【請求項 1】 第 1 の時間情報を外部から受信する受信手段と、

第 2 の時間情報を生成する時間情報生成手段と、  
前記第 1 の時間情報と前記第 2 の時間情報との所定以上の時間差を検出する時間差検出手段と、  
前記所定以上の時間差に応じて前記時間情報生成手段が生成する第 2 の時間情報を修正する修正手段とを有する通信装置。

【請求項 2】 前記受信手段は、前記第 1 の時間情報とそれに対応するデータを受信する手段であり、さらに、前記受信手段が受信するデータを蓄積するためのバッファと、

前記第 1 の時間情報を受信した時から所定時間経過後に前記バッファに蓄積されている該第 1 の時間情報に対応するデータの処理を開始するデータ処理手段とを有し、前記データ処理手段は、受信した前記第 1 の時間情報と前記第 2 の時間情報との時間差が所定値を超えてから前記バッファに蓄積されている該第 1 の時間情報に対応するデータの処理を開始する請求項 1 記載の通信装置。

【請求項 3】 (a) 第 1 の時間情報を外部から受信する工程と、

(b) 第 2 の時間情報を生成する工程と、

(c) 前記第 1 の時間情報と前記第 2 の時間情報との所定以上の時間差を検出する工程と、

(d) 前記所定以上の時間差に応じて前記第 2 の時間情報を修正する工程とを有する通信方法。

【請求項 4】 (a) 第 1 の時間情報を外部から受信する手順と、

(b) 第 2 の時間情報を生成する手順と、

(c) 前記第 1 の時間情報と前記第 2 の時間情報との所定以上の時間差を検出する手順と、

(d) 前記所定以上の時間差に応じて前記第 2 の時間情報を修正する手順とをコンピュータに実行させるためのプログラムを記録した媒体。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、通信技術に関し、特に時間情報を通信する通信技術に関する。

## 【0002】

【従来の技術】電子楽器間の通信の統一規格として、MIDI (musical instrument digital interface) 規格がある。MIDI 規格のインターフェースを備えた電子楽器は、MIDI 用ケーブルを用いて、他の電子楽器と接続することができる。電子楽器は、MIDI 用ケーブルを介して、MIDI データを通信することができる。例えば、一つの電子楽器は、演奏者が演奏した情報を MIDI データとして送信し、他の電子楽器は、当該 MIDI データを受信し、楽音を発音することができる。一つの電子楽器で演奏すると、他の電子楽器でリアルタイムに発音することができる。

ムに発音することができる。

【0003】また、複数の汎用コンピュータを接続する通信ネットワークでは、種々の情報を通信することができる。例えば、コンピュータに接続されているハードディスク等にオーディオデータ (生の楽音情報) や MIDI データ等の情報を一度蓄積しておき、通信ネットワークを介して、当該情報を送信する。他のコンピュータは、当該情報を受信して、ハードディスク等の記憶装置に記憶する。汎用の通信ネットワークは、情報の通信を行うのみであり、MIDI とは性質を異にする。

【0004】MIDI 規格は、電子楽器間のリアルタイム通信を可能にするが、長距離の通信及び多数ノード間の通信に適していない。一方、汎用通信ネットワークは、長距離の通信及び多数ノード間の通信に適しているが、電子楽器間のリアルタイム通信を考慮したものではない。

## 【0005】

【発明が解決しようとする課題】汎用ネットワークにおいて、時間情報を通信する場合を考える。送信装置と受信装置は、汎用ネットワークに接続され、それぞれが独自のタイマを有し、日付や時刻の時間情報を生成することができる。送信装置のタイマと受信装置のタイマは、通常、同期がとれていないので、両装置が生成する時間情報は必ずしも一致せず、ずれることがある。

【0006】送信装置が生成した時間情報を受信装置に送信すると、その時間情報は、受信装置が生成した時間情報と必ずしも一致していない。両者の時間情報が一致していない場合に、受信装置が、その時間情報に応じて、データの処理を行うと、種々の不都合が生じることがある。

【0007】また、汎用ネットワークは、長距離通信を行ったり、その通信経路が種々変化する場合がありますので、各パケット毎の通信時間は一定とは限らない。この通信時間の変化は、受信装置が送信装置から受信した時間情報に応じて処理をする際に悪影響を及ぼすことがある。

【0008】本発明の目的は、時間情報を通信する際に時間情報の信頼性を高めることができる通信装置、通信方法又はプログラムを記録した媒体を提供することである。

## 【0009】

【課題を解決するための手段】本発明の一観点によれば、第 1 の時間情報を外部から受信する受信手段と、第 2 の時間情報を生成する時間情報生成手段と、前記第 1 の時間情報と前記第 2 の時間情報との所定以上の時間差を検出する時間差検出手段と、前記所定以上の時間差に応じて前記時間情報生成手段が生成する第 2 の時間情報を修正する修正手段とを有する通信装置が提供される。

【0010】第 1 の時間情報は、外部の送信装置により生成され、送信されたものを受信する。第 2 の時間情報

は、時間情報生成手段が生成する。第1の時間情報と第2の時間情報とは、必ずしも同期がとれていない。第1の時間情報と第2の時間情報との時間差を検出し、その時間差に応じて第2の時間情報を修正することにより、第1の時間情報と第2の時間情報とのずれを修復することができる。

#### 【0011】

【発明の実施の形態】図2は、楽音情報及び画像情報の通信ネットワークを示す図である。

【0012】演奏会場1には、MIDI楽器2、音声入力装置12、カメラ4、エンコーダー3、5、及びルータ6が備えられる。音声入力装置12は、例えばマイクロフォンである。演奏会場1では、演奏者がMIDI楽器2を演奏し、歌手がその演奏にあわせて音声入力装置12に向かって歌う。また、音声入力装置12を生ドラムや生ピアノやエレキギターのそばに置いて、それらの音を音声入力装置12に入力してもよい。

【0013】MIDI楽器2は、演奏者の演奏操作に応じてMIDIデータを生成し、エンコーダー3に供給する。音声入力装置12は、歌手の歌声又はドラムの音等を電気信号に変換してアナログ形式のオーディオ信号（音声信号）を生成し、リアルタイムでエンコーダー3に供給する。エンコーダー3は、アナログ形式のオーディオ信号をデジタル形式のオーディオデータ（音声データ）に変換し、MIDIデータ及びオーディオデータを所定のデータ形式で、ルータ6を介してインターネット上にパケット送信する。データ形式は、後に図4

(A)、(B)を参照しながら説明する。

【0014】カメラ4は、演奏者が演奏している様子を撮影し、その様子を画像データとしてエンコーダー5に供給する。エンコーダー5は、画像データを所定のデータ形式で、ルータ6を介してインターネット上にパケット送信する。

【0015】ルータ6は、以下に示すインターネットを介して、MIDIデータ、オーディオデータ及び画像データを送信する。当該データは、電話回線又は専用回線を通り、ルータ6からサーバー7に供給され、さらに複数のWWW (world wide web) サーバー8に供給される。WWWサーバー8は、いわゆるプロバイダである。

【0016】ユーザは、ホームコンピュータ9をWWWサーバー8に接続することにより、インターネットを使用することができる。ホームコンピュータ9は、インターネットを使用し、MIDIデータ、オーディオデータ及び画像データを受信することができる。ホームコンピュータ9は、ディスプレイ装置とMIDI音源を有し、MIDI音源は音声出力装置11に接続される。

【0017】画像データは、ディスプレイ装置に表示される。MIDIデータは、MIDI音源で楽音信号に変換され、音声出力装置11で発音される。オーディオデータは、デジタル形式からアナログ形式に変換されて、

音声出力装置11で発音される。ホームコンピュータ9は、MIDIデータとオーディオデータの同期をとって、両データに応じて音声出力装置11で発音させる。演奏会場1の演奏音や歌声と同等の音が音声出力装置11からリアルタイムで発音される。

【0018】また、ホームコンピュータ9の外部に、MIDI音源10を接続すれば、ホームコンピュータ9は、MIDI音源10に楽音信号を生成させ、音声出力装置11から発音させることができる。

【0019】なお、音楽の演奏において、ユーザにとっては、画像データよりもMIDIデータ及びオーディオデータの方が重要な情報であるので、画像データよりもMIDIデータ及びオーディオデータを優先して処理を行う。画像データは、画質が悪く、コマ数が少なくてもさほど気にならないが、MIDIデータ及びオーディオデータに基づく楽音信号は高品質が要求される。なお、スポーツの中継等の場合は、画像と音との関係が逆になる。

【0020】ユーザは、演奏会場1に赴かなくても、自宅にいながらディスプレイ装置で演奏会場1の様態を見ながら、リアルタイムで演奏音及び歌声を聴くことができる。また、自宅のホームコンピュータ9をインターネットに接続すれば、誰でも演奏音及び歌声を聴くことができる。例えば、演奏会場1でコンサートを行った場合には、不特定多数人が自宅でそのコンサートを楽しむことができる。

【0021】演奏会場からMIDIデータを自宅に送信することにより、演奏者が複数のユーザのそれぞれの自宅で電子楽器を演奏しているかのような状況を作り出すことができる。MIDIデータの通信は、雑音により音質を下げることはない。

【0022】図3は、送信端であるエンコーダー3と受信端であるホームコンピュータ9を示す。以下、両者の関係を説明するため、便宜的にエンコーダー3をサーバ3と呼び、ホームコンピュータ9をクライアント9と呼ぶ。

【0023】サーバ3とクライアント9は、インターネットのデジタル回線により接続される。サーバ3は、MIDI楽器2からMIDIデータを受け、音声入力装置12からアナログ形式のオーディオ信号を受ける。デジタル形式に変換されたオーディオデータ及びMIDIデータは、サーバ3からクライアント9に送信される。クライアント9は、MIDI音源10にMIDIデータを出力し、音声出力装置11にアナログ形式に変換されたオーディオ信号を出力する。MIDI音源10は、MIDIデータを受け、アナログ形式の楽音信号を生成して音声出力装置11に出力する。音声出力装置11は、アナログ形式のオーディオ信号及び楽音信号を受けて発音する。

【0024】図4(A)は、サーバ3が送信するMIDI

Iデータパケット49の構造を示す。

【0025】MIDIデータパケット49は、ヘッダ51とMIDIデータ44とフッタ52を有する。ヘッダ51は、時間情報を表すタイムスタンプ41、パケットの順番を示すシーケンスナンバ53、当該パケットがMIDIデータであることを示す識別子(ID)42、当該パケットのサイズ43を有する。

【0026】タイムスタンプ41は、パケット内のMIDIデータ44の送信時刻を表すと共に、演奏時刻、録音時刻、再生時刻をも表す。サーバ3は、自己のシステムクロック(タイマ)が生成する時間情報に応じて、タイムスタンプ41を生成する。

【0027】シーケンスナンバ53は、パケット単位で付与される順番を表す番号である。クライアント9は、通信エラーによりデータの順序が入れ替わった場合でも、パケットを所定時間だけバッファリングした後、バッファ内のデータをシーケンスナンバ53に従ってソート処理することにより、通信エラーをリカバリすることができる。

【0028】識別子42は、MIDIデータパケットやオーディオデータパケットや画像データパケット等のパケットの種類を表すことが可能である。ここでは、MIDIデータ44を送信するので、MIDIデータパケットを表すものとなる。

【0029】MIDIデータ44は、スタンダードMIDIファイルフォーマットに準拠したものであり、デルタタイム(インターバル)とMIDIイベントを1組にしたデータの列である。デルタタイムは、直前のMIDIイベントと当該MIDIイベントの間の時間間隔を表す。ただし、デルタタイムが0であるときには、デルタタイムを省略することができる。

【0030】フッタ52は、データの終了を表すデータを有する。ヘッダ51又はフッタ52に、チェックサムを含ませてもよい。チェックサムは、例えばMIDIデータ44の合計値である。この場合、サーバ3は、当該合計値を計算し、チェックサムとしてパケット中に付与する。クライアント9は、当該合計値を計算し、チェックサムの値と合っていれば、通信エラーがないことを確認することができる。

【0031】図4(B)は、サーバ3が送信するオーディオデータパケット50の構造を示す。

【0032】オーディオデータパケット50は、ヘッダ51とデジタルオーディオデータ48とフッタ52を有する。デジタルオーディオデータ48は、音声入力装置12(図3)から生成されたデータを、A/D変換し、圧縮したデータである。

【0033】ヘッダ51は、MIDIデータパケットの場合と同様に、タイムスタンプ41、シーケンスナンバ53、当該パケットがオーディオデータであることを示す識別子(ID)42、当該パケットのサイズ43を有

する。タイムスタンプ41は、MIDIデータパケットの場合と同様に、パケット内のオーディオデータ48の送信時刻等を表す。

【0034】次に、タイムスタンプと通信時間との関係を説明する。パケット内には、タイムスタンプが含まれている。クライアント9は、このタイムスタンプに応じて、MIDIデータ等の処理を行えばよい。

【0035】しかし、インターネット等では、パケット毎に通信時間が異なる場合がある。すなわち、あるパケットは通信時間が長く、他のパケットは通信時間が短い場合がある。これは、長距離通信の場合や通信経路がパケット毎に変化する場合に顕著に現れる。

【0036】また、通信エラーにより、送信時のパケットの順番と受信時のパケットの順番が異なることがある。すなわち、パケットの通信順序の入れ替えが発生することがある。

【0037】クライアント9は、受信データをバッファにバッファリングし、タイムスタンプの時刻から所定時間(例えば3秒)経過後に、バッファ内のデータの処理を開始する。

【0038】バッファ内にデータをバッファリングすることにより、パケット毎の通信時間の相違を吸収することができる。また、上記のシーケンスナンバ53(図4(A)、(B))に応じて、パケット内のデータをソート処理することにより、パケットの順序を正常なものに戻すことができる。

【0039】次に、サーバ3のシステムクロックとクライアント9のシステムクロックとのずれによる影響を説明する。サーバ3のシステムクロックとクライアント9のシステムクロックは、それぞれ独自に時間情報を生成し、両者は同期がとれていない。したがって、両者のシステムクロックが生成する時間情報は時間経過と共にずれていくことがある。これは、一般の時計に誤差があるのと同じである。

【0040】例えば、クライアント9のシステムクロックがサーバ3のものよりも進んだ場合を考える。当初は、3秒経過後にバッファ内のデータを取り出しているも、徐々にその時間が短くなり、やがては受信したデータを直に取り出して、処理するようになってしまう。この場合、一定時間だけバッファリングする機能を失うことになる。

【0041】次に、クライアント9のシステムクロックがサーバ3のものよりも遅れた場合を考える。当初は、3秒経過後にバッファ内のデータを取り出しているも、徐々にその時間が長くなり、時間の経過と共にバッファ内のデータ量が増加して行く。バッファの容量が小さい場合には、バッファからデータが溢れて、データを失う事態も起こりえる。

【0042】以上の弊害を除去するため、クライアント9のシステムクロックをサーバ3のものに合わせる(同

期させる)ための修正を行う。具体的には、クライアント9のシステムクロックが生成するタイムスタンプを、サーバ3のシステムクロックが生成するタイムスタンプに一致させる。その詳細を、以下説明する。

【0043】図1(A)は、サーバのタイムスタンプとクライアントのタイムスタンプがずれる過程を示すタイムチャートである。横軸は、時間を示す。

【0044】サーバは、送信開始時に、タイムスタンプを0分にして、そのタイムスタンプを含んだパケットP0をクライアントに送信する。サーバのシステムクロックは、その後、そのタイムスタンプを順次カウントアップする。

【0045】クライアントは、0分のタイムスタンプを含むパケットP0を受信すると、その時点で、クライアントのタイムスタンプを0分にリセットする。クライアントのシステムクロックは、その後、当該タイムスタンプを順次カウントアップする。サーバのタイムスタンプとクライアントのタイムスタンプは、同期がとれていない。

【0046】ここで、クライアントのタイムスタンプがサーバのものよりも進む場合を考える。例えば、サーバのタイムスタンプが15分の時、クライアントのタイムスタンプが15分300m秒であるとする。クライアントのタイムスタンプの進みは、300m秒である。

【0047】サーバが、15分経過後に、15分のタイムスタンプを含むパケットP1をクライアントに送信するとする。クライアントは、パケットP1を受信した時点で、両者のタイムスタンプを比較し、自己のタイムスタンプが300m秒進んでいることを認識することができる。

【0048】図1(B)は、タイムスタンプの修正方法を示すタイムチャートである。横軸は時間である。図1(A)の15分の時点を時刻T1とする。

【0049】時刻T1の時点で、クライアントのタイムスタンプは、サーバのものよりも300m秒進んでいる。クライアントは、自己のタイムスタンプが例えば300m秒以上進んだことを所定データ数以上認識した時点で、タイムスタンプの修正処理を開始する。この際、タイムスタンプのずれを一気に修正するのではなく、以下の方法により、徐々に修正していく。

【0050】時刻T1では、300m秒のずれがある。時刻T1から、5秒経過後の時刻T2までの間に100m秒だけクライアントのタイムスタンプを修正する。時刻T2では、200m秒のずれがある。続いて、時刻T2から、5秒経過後の時刻T3までの間に再び100m秒修正する。時刻T3では、100m秒のずれがある。時刻T3から、5秒経過後の時刻T4までの間に再び100m秒修正する。時刻T4では、ずれがなくなる。以上で、タイムスタンプの修正処理が終了する。タイムスタンプの修正は、約15秒かけて、100m秒ずつ3回

に分けて行う。

【0051】図5は、図3の具体的なハードウェア構成を示す図である。サーバ3とクライアント9は、共に汎用コンピュータ又はパーソナルコンピュータ等を用いることができる。

【0052】サーバ3とクライアント9は、基本的に同じ構成である。両者の構成を説明する。バス21には、CPU22、RAM24、外部記憶装置25、外部に対してMIDIデータを送受信するためのMIDIインターフェース26、サウンドカード27、ROM28、表示装置29、キーボードやスイッチやマウス等の入力手段30、インターネットを行うための通信インターフェース31が接続されている。

【0053】サウンドカード27は、バッファ27aとコーデック回路27bを有する。バッファ27は、外部に対して入力又は出力するためのデータをバッファリングする。コーデック回路27bは、A/D変換器及びD/A変換器を有し、アナログ形式とデジタル形式の両者間の変換を行うことができる。さらに、コーデック回路27bは、圧縮/伸張回路を有し、データの圧縮及び伸張を行うことができる。データは、圧縮された状態で、インターネット通信される。

【0054】外部記憶装置25は、例えばハードディスクドライブ、フロッピーディスクドライブ、CD-ROMドライブ、光磁気ディスクドライブ等であり、MIDIデータ、オーディオデータ、画像データ又はコンピュータプログラム等を記憶することができる。

【0055】ROM28は、コンピュータプログラム及び各種パラメータを記憶することができる。RAM24は、バッファやレジスタ等のワーキングエリアを有し、外部記憶装置25に記憶されている内容をコピーして記憶することができる。

【0056】CPU22は、ROM28又はRAM24に記憶されているコンピュータプログラムに従って、各種演算または処理を行う。CPU22は、システムクロック23から時間情報を得て、タイマ割り込み処理を行うことができる。サーバ3のシステムクロック23は、サーバのタイムスタンプを生成し、クライアント9のシステムクロック23は、クライアントのタイムスタンプを生成する。

【0057】インターネット回線32には、サーバ3の通信インターフェース31及びクライアント9の通信インターフェース31が接続される。通信インターフェース31は、インターネットにより、MIDIデータ、オーディオデータ及び画像データを送受信するためのインターフェースである。サーバ3とクライアント9は、インターネット回線32により接続される。

【0058】まず、サーバ3について説明する。MIDIインターフェース26には、MIDI楽器2が接続され、サウンドカード27には、音声入力装置12が接続

される。MIDI 楽器 2 は、演奏者の演奏操作に応じて MIDI データを生成し、MIDI インタフェース 26 に出力する。音声入力装置 12 は、演奏会場における音声を入力し、アナログ形式のオーディオ信号をサウンドカード 27 に出力する。サウンドカード 27 は、アナログ形式のオーディオ信号をバッファ 27 a にバッファリングし、コーデック回路 27 b でアナログ形式のオーディオ信号をデジタル形式のオーディオデータに変換し、そのデータを圧縮する。

【0059】次に、クライアント 9 について説明する。図 3 に示すように、MIDI インタフェース 26 には、MIDI 音源 10 が接続され、サウンドカード 27 には、音声出力装置 11 が接続される。CPU 22 は、通信インタフェース 31 を介して、インターネット回線 32 上から MIDI データとオーディオデータと画像データを受信する。

【0060】図 6 に示すように、クライアント 9 の RAM 24 は、受信バッファ 24 a、パケット内のタイムスタンプ（サーバのタイムスタンプ）を格納するレジスタ 24 b、自己のタイムスタンプ（クライアントのタイムスタンプ）を格納するレジスタ 24 c、タイムスタンプのずれの大きさを表すずれ検出カウンタ値を格納するレジスタ 24 d を有する。

【0061】通信インタフェース 31 は、インターネット用インタフェースの他、イーサネット用インタフェース、IEEE 1394 規格のデジタル通信インタフェース、RS-232C 用インタフェースでもよく、種々のネットワークに接続することができる。

【0062】サーバ 3 は、MIDI データ等を送信するためのコンピュータプログラムを記憶する。クライアント 9 は、MIDI データ等を受信したり、タイムスタンプを修正するためのコンピュータプログラムを記憶する。コンピュータプログラムや各種パラメータ等を外部記憶装置 25 に記憶させておき、それを RAM 24 に読み込むことにより、コンピュータプログラム等の追加やバージョンアップ等が容易に行える。

【0063】CD-ROM（コンパクトディスクリード・オンリィ・メモリ）ドライブは、CD-ROM に記憶されているコンピュータプログラム等を読み出す装置である。読み出したコンピュータプログラム等は、ハードディスクにストアされる。コンピュータプログラムの新規インストールやバージョンアップ等が容易に行える。

【0064】通信インタフェース 31 は LAN（ローカルエリアネットワーク）やインターネット、電話回路等の通信ネットワーク 32 に接続されており、該通信ネットワーク 32 を介して、コンピュータ 33 と接続される。外部記憶装置 25 内に上記のコンピュータプログラム等が記憶されていない場合、コンピュータ 33 からコンピュータプログラム等をダウンロードすることができ

る。サーバ 3 又はクライアント 9 は、通信インタフェース 31 及び通信ネットワーク 32 を介してコンピュータ 33 へコンピュータプログラム等のダウンロードを要求するコマンドを送信する。コンピュータ 33 は、このコマンドを受け、要求されたコンピュータプログラム等を、通信ネットワーク 32 を介してサーバ 3 又はクライアント 9 へ配信する。サーバ 3 又はクライアント 9 が通信インタフェース 31 を介して、コンピュータプログラム等を受信して外部記憶装置 25 に蓄積することにより、ダウンロードが完了する。

【0065】なお、本実施例は、本実施例に対応するコンピュータプログラム等をインストールした市販のパーソナルコンピュータ等によって、実施させるようにしてもよい。その場合には、本実施例に対応するコンピュータプログラム等を、CD-ROM やフロッピディスク等の、コンピュータが読み込むことができる記憶媒体に記憶させた状態で、ユーザーに提供してもよい。そのパーソナルコンピュータ等が、LAN、インターネット、電話回線等の通信ネットワークに接続されている場合には、通信ネットワークを介して、コンピュータプログラムや各種データ等をパーソナルコンピュータ等に提供してもよい。

【0066】また、サーバ 3 又はクライアント 9 は、パーソナルコンピュータの他、電子楽器、ゲーム機、カラオケ装置、テレビ等の形態として適用してもよい。

【0067】図 7～図 9 に、サーバが行う処理のフローチャートを示す。図 7 は、サーバが行うオーディオデータの送信処理を示すフローチャートである。

【0068】ステップ SA1 では、音声入力装置 12（図 2）からオーディオデータを受信する。

【0069】ステップ SA2 では、受信したオーディオデータをアナログ形式からデジタル形式に変換する。具体的には、所定のサンプリングレートで、オーディオデータのサンプリングを行い、デジタル形式のオーディオデータを生成する。

【0070】ステップ SA3 では、デジタル形式のオーディオデータを圧縮し、データ量を削減する。ステップ SA2 の A/D 変換及びステップ SA3 の圧縮処理は、コーデック回路 27 b（図 5）で行うことができる。

【0071】ステップ SA4 では、圧縮したデータに、タイムスタンプを付与して、パケット化する。このパケットは、図 4（B）に示す構造である。タイムスタンプの生成方法は、後に図 9 を参照しながら説明する。このタイムスタンプは、サーバのタイムスタンプである。

【0072】ステップ SA5 では、当該パケットをインターネットにより送信する。以上で、オーディオデータの送信処理が終了する。

【0073】図 8 は、サーバが行う MIDI データの送信処理を示すフローチャートである。

【0074】ステップ SB1 では、MIDI 楽器 2（図

2) からMIDIデータを受信する。

【0075】ステップSB2では、受信したMIDIデータに、タイムスタンプを付与して、パケット化する。このパケットは、図4(A)に示す構造である。タイムスタンプの生成方法は、後に図9を参照しながら説明する。このタイムスタンプも、サーバのタイムスタンプである。

【0076】ステップSB3では、当該パケットをインターネットにより送信する。以上で、MIDIデータの送信処理が終了する。

【0077】図9は、サーバがタイムスタンプを生成する処理を示すフローチャートである。この処理は、上記のパケットを送信する際に必要とされる処理であり、必ずしもタイマ割り込みにより定期的に行う必要はない。

【0078】ステップSC1では、演奏開始時における最初のパケット作成であるか否か、すなわち、最初に送信するパケットであるか否かをチェックする。最初のパケットであるときには、yesの矢印に従い、ステップSC2へ進む。最初のパケットでないときには、noの矢印に従い、ステップSC4へ進む。

【0079】ステップSC2では、サーバのシステムクロックの値を読み出し、記憶部(RAM24)に演奏開始時刻として記憶する。

【0080】ステップSC3では、サーバのタイムスタンプの値を0にクリアする。すなわち、タイムスタンプは、演奏開始時(送信開始時)に0に設定される。以上により、タイムスタンプ生成の処理を終了する。

【0081】次に、パケットが最初のパケットでない場合を説明する。この場合、上記のように、ステップSC1からステップSC4へ進む。

【0082】ステップSC4では、サーバのシステムクロックの値を読み出す。この値は、パケットの送信時刻に相当する。

【0083】ステップSC5では、記憶部(例えばRAM24)に記憶されている演奏開始時刻を読み出す。演奏開始時刻は、上記のステップSC2にて、既に記録されている。

【0084】ステップSC6では、ステップSC4で読み出したシステムクロックの値から、ステップSC5で読み出した演奏開始時刻を減算し、その値をタイムスタンプとする。タイムスタンプは、演奏開始時を0としたときの時間情報であり、サーバのシステムクロックによりカウントアップされる。以上により、タイムスタンプ生成の処理を終了する。なお、タイムスタンプは、タイマ割り込みによるカウンタにより生成してもよい。

【0085】図10～図14に、クライアントが行う処理のフローチャートを示す。図10は、クライアントの受信処理を示すフローチャートである。

【0086】ステップSD1では、パケットを受信する。ステップSD2では、受信したパケットが最初のパ

ケットであるか否かを判断し、最初のパケットであれば、クライアントのタイムスタンプの値を0にクリアする。最初のパケット内には、サーバのタイムスタンプとして0が含まれている。パケット内のタイムスタンプとクライアントのタイムスタンプは、共に0になる。

【0087】図11に示すように、クライアントのタイムスタンプは、クライアントのタイマ割り込み処理により生成される。ステップSE1では、クライアントのシステムクロックに同期してタイムスタンプをカウントアップする。タイムスタンプは、システムクロックと同じタイミングで増加する。なお、クライアントのタイムスタンプは、図9のサーバのタイムスタンプ生成処理と同様な方法により生成してもよい。

【0088】図10に戻り、ステップSD3では、パケット内のタイムスタンプを取得する。パケット内のタイムスタンプは、サーバが生成するものであり、演奏開始時からの時間に相当する。当該パケットが最初のパケットであれば、タイムスタンプは0である。

【0089】ステップSD4では、クライアントのタイムスタンプを取得する。このタイムスタンプは、上記の図11の処理によりカウントアップされる。

【0090】ステップSD5では、ステップSD3で取得したパケット内のタイムスタンプとステップSD4で取得したクライアントのタイムスタンプを比較し、両者のずれ量を求める。

【0091】ステップSD6では、ずれ量が許容範囲内であるか否かをチェックする。クライアントとサーバのシステムクロックにずれがない場合にも、通信状況により、わずかなタイムスタンプのずれが生じることがあるので、そのようなずれを除外するため、上記のずれ量に許容範囲を設けている。この許容範囲(設定修正時間)に相当する値はクライアント側で任意に設定できるものであっても構わない。

【0092】ずれ量が許容範囲を超えているときには、noの矢印に従い、ステップSD8へ進む。ずれ量が許容範囲内であるときには、yesの矢印に従い、ステップSD7へ進む。

【0093】ステップSD8では、ずれ量が許容範囲を超えているので、ずれ検出カウンタ24d(図6)の値を以下のように調整する。クライアントのタイムスタンプが遅れていれば、ずれ検出カウンタの値を増加させる。逆に、クライアントのタイムスタンプが進んでいれば、ずれ検出カウンタの値を減少させる。ずれ検出カウンタの値は、初期時が0であり、正值であれば遅れを示し、負値であれば進みを示す。

【0094】パケットを受信する度に、タイムスタンプのずれ量が判別され、ずれ検出カウンタがカウントを行う。ずれ検出カウンタは、許容範囲(設定修正時間)を越えたずれの回数(累算値)を示す。ずれ検出カウンタの値が所定値を超えると、後に示す図13の処理でタイ



ムスタンプが修正されることになる。その後、ステップSD9へ進む。

【0095】一方、ステップSD7では、ずれ量が許容範囲内であるので、ずれ検出カウンタの絶対値をカウントダウンする。例えば、ずれ検出カウンタが5であれば4にし、-5であれば-4にする。

【0096】タイムスタンプのずれは一時的な場合がある。つまり、前回はずれが発生したが、今回はずれが発生しない場合がある。その場合は、タイムスタンプのずれがなくなったと判断し、ずれ検出カウンタの絶対値をカウントダウンする。この場合、ずれ検出カウンタは、0にリミットされる。その後、ステップSD9へ進む。

【0097】ステップSD9では、上記で受信したパケットをデータの種類に応じて、異なるバッファにバッファリングする。データの種類は、オーディオデータ、MIDIデータ、画像データであり、データID42(図4(A)、(B))により判断される。以上で、クライアントの受信処理が終了する。

【0098】図12は、クライアントが行う再生処理を示すフローチャートである。ステップSF1では、データの種類毎のバッファ(例えばMIDIデータ用バッファやオーディオデータ用バッファ等)を定期的に検索し、各バッファ内にデータが存在するか否かをチェックする。存在するときには、そのデータをバッファから読み出す。

【0099】ステップSF2では、現在のクライアントのタイムスタンプを取得する。当該タイムスタンプは、上記の図11の処理でカウントアップされる。

【0100】ステップSF3では、ステップSF1で読み出したデータに対応するパケット内のタイムスタンプ(サーバのタイムスタンプ)を取得する。

【0101】ステップSF4では、ステップSF2で取得したクライアントのタイムスタンプとステップSF3で取得したパケット内のタイムスタンプを比較し、両者の差をディレイ時間として求める。

【0102】ステップSF5では、当該ディレイ時間が所定値に達したか否かをチェックする。このディレイ時間は、バッファ内にデータを蓄積する時間に相当し、例えば3秒である。

【0103】ディレイ時間が所定値に達したときには、yesの矢印に従い、ステップSF6へ進み、当該データを再生する処理を行う。ディレイ時間が所定値に達していないときには、noの矢印に従い、処理を終了し、ディレイ時間が所定値に達するまでデータの再生を行わない。以上により、クライアントの再生処理を終了する。

【0104】上記のように、一定時間(例えば3秒)だけデータをバッファ内に蓄積し、その後にデータの再生を行うことにより、各パケット毎の通信時間の相違(ずれ)やパケットの通信順序の入れ替えによる悪影響を防

止することができる。

【0105】図13は、クライアントのタイムスタンプのずれを検出する処理を示すフローチャートである。このフローに相当する処理は定期的に起動されても良いし、図10のSD8の処理後SD9の前に挿入して起動されても構わない。

【0106】ステップSG1では、ずれ検出カウンタが所定値を超えたか否かをチェックする。ずれ検出カウンタは、許容範囲(設定修正時間)を越えたずれの回数(累算)をカウントするものであり、図10のステップSD8でカウントされる。ずれの回数が所定値を超えていれば、そのずれは一時的なずれではなく、タイムスタンプを修正する必要がある。

【0107】所定値を超えたときには、yesの矢印に従い、ステップSG2へ進み、タイムスタンプのずれの修正の準備を行う。一方、所定値を超えないときには、ずれが一時的なものであるとして、タイムスタンプのずれの修正を行わずに、noの矢印に従い、処理を終了する。

【0108】ステップSG2では、ずれ検出カウンタ値をタイムスタンプ補正モジュールへ引き渡し、ずれ検出カウンタをクリアする。タイムスタンプ補正モジュールは、後に示す図14の処理に相当する。

【0109】ステップSG3では、タイムスタンプ補正モジュールに起動がかかるように処理する。例えば、起動用フラグを用いて、このフラグがセットされたときに当該モジュールが起動するようにしてもよいし、当該モジュールを直接呼び出(コール)してもよい。以上により、タイムスタンプのずれ検出処理が終了する。

【0110】図14は、クライアントのタイムスタンプ補正モジュールの処理を示すフローチャートである。ステップSH1では、ステップSG2で引数として渡されたずれ検出カウンタ値を取得する。

【0111】ステップSH2では、ずれ検出カウンタ値を基に、クライアントのタイムスタンプが遅れているか又は進んでいるかを判断する。例えば、ずれ検出カウンタ値が正值であれば遅れであり、負値であれば進みである。

【0112】次に、遅れか進みかにより、クライアントのタイムスタンプを修正する時間方向を決定する。クライアントのタイムスタンプが遅れているときには、前記クライアントのタイムスタンプを加算する方向(進む方向)に修正し、進んでいる時には減算する方向(遅らせる方向)に修正するように決定する。

【0113】ステップSH3では、設定修正時間に応じて、補正值をテーブル等から取得し、補正值レジスタにセットする。例えば、前記設定修正時間が300m秒であれば、補正值レジスタに3をセットする。

【0114】ステップSH4では、補正值レジスタの値が0であるか否かをチェックする。0でないときには、

noの矢印に従い、ステップSH5へ進む。

【0115】ステップSH5では、所定値をタイムスタンプに加算又は減算する。所定値は、補正値の1目盛りに対応する修正時間であり、例えば100m秒(図1(B))である。加算又は減算は、上記のステップSH2の決定に従い、いずれかが行われる。

【0116】ステップSH6では、補正値から所定値を減算する。所定値は、例えば1である。その後、所定時間経過後に、ステップSH4へ戻る。所定時間は、例えば5秒(図1(B))である。

【0117】ステップSH4では、再び補正値レジスタが0か否かをチェックする。0でなければ、上記のステップSH5及びSH6の処理を繰り返す。補正値レジスタが0になれば、処理を終了する。

【0118】補正値レジスタが0になるまで、ステップSH4、SH5、SH6を含むステップSH10の処理は、定期的(例えば5秒間隔)に行われる。

【0119】以上のように、タイムスタンプが所定値(例えば300m秒)以上ずれていることが所定回数以上検出されると、所定時間(例えば5秒)おきに所定時間(例えば100m秒)ずつタイムスタンプを修正する。300m秒のずれを一度に修正するのではなく、15秒の時間をかけて徐々に100m秒ずつ3回に分けてタイムスタンプのずれを滑らかに修正する。これにより、データの再生速度が急激に変化することを防止し、滑らかな再生を行うことができる。

【0120】パケットのタイムスタンプとクライアントのタイムスタンプとの差が所定値を超えたときに再生処理により発音されるので(図12のステップSF5、SF6)、上記のタイムスタンプの修正を行っても、途中の音の発音が省略されることはない。

【0121】本実施例によれば、クライアントは、受信したデータをバッファ内に所定時間(例えば3秒)だけ格納した後に、データの再生処理を行う。これにより、各パケット毎に通信時間が異なる場合やパケットの通信順序が入れ替わった場合にも、適切な再生時間かつ適切なデータ順序で再生することができる。

【0122】クライアントのタイムスタンプが進んだ場合には、バッファ内にデータを蓄積(バッファリング)する時間が次第に短くなり、最終的には蓄積する時間が0になってしまい、データをバッファリングする上記の機能が失われてしまう。

【0123】逆に、クライアントのタイムスタンプが遅れた場合には、バッファ内に蓄積するデータ量が次第に増加する。バッファ容量が少ない場合には、やがてバッファ内のデータがあふれてしまい、データが失われることがある。

【0124】クライアントは、クライアントのタイムスタンプがずれた(進んだり遅れた)ことを検出した場合には、そのタイムスタンプを修正することができる。こ

れにより、上記の問題点を解消することができる。その際、ずれが大きいときには、一度に修正するのではなく、複数回に分けて少しずつ修正する。これにより、タイムスタンプを修正する際にも、データを滑らかに再生することができる。

【0125】なお、本実施例は、オーディオデータ及びMIDIデータ等をインターネットで通信する場合に限定されない。例えば、IEEE1394規格のデジタルシリアル通信や通信衛星等の他の通信にも適用することができる。

【0126】以上実施例に沿って本発明を説明したが、本発明はこれらに制限されるものではない。例えば、種々の変更、改良、組み合わせ等が可能なことは当業者に自明であろう。

【0127】

【発明の効果】以上説明したように、本発明によれば、外部から受信する第1の時間情報と自己が生成する第2の時間情報との同期がとれていなくても、第1の時間情報と第2の時間情報との時間差を検出し、その時間差に応じて第2の時間情報を修正することができる。第1の時間情報と第2の時間情報との間にずれが発生しても、そのずれを修正することができるので、そのずれによる弊害を防止することができる。

【図面の簡単な説明】

【図1】 図1(A)はタイムスタンプのずれの発生を示すタイムチャートであり、図1(B)はタイムスタンプの修正を示すタイムチャートである。

【図2】 楽音情報及び画像情報の通信ネットワークを示す図である。

【図3】 送信端であるエンコーダー(サーバ)と受信端であるホームコンピュータ(クライアント)の接続を示す図である。

【図4】 図4(A)はMIDIデータパケットの構造を示し、図4(B)はオーディオデータパケットの構造を示す図である。

【図5】 サーバ及びクライアントのハードウェアの構成を示す図である。

【図6】 クライアントのRAMを示す図である。

【図7】 サーバが行うオーディオデータの処理を示すフローチャートである。

【図8】 サーバが行うMIDIデータの処理を示すフローチャートである。

【図9】 サーバが行うタイムスタンプの生成処理を示すフローチャートである。

【図10】 クライアントが行う受信処理を示すフローチャートである。

【図11】 クライアントが行うタイムスタンプの生成処理を示すフローチャートである。

【図12】 クライアントが行う再生処理を示すフローチャートである。

17

【図13】 クライアントが行うタイムスタンプのずれ検出処理を示すフローチャートである。

【図14】 クライアントが行うタイムスタンプ補正モジュールの処理を示すフローチャートである。

【符号の説明】

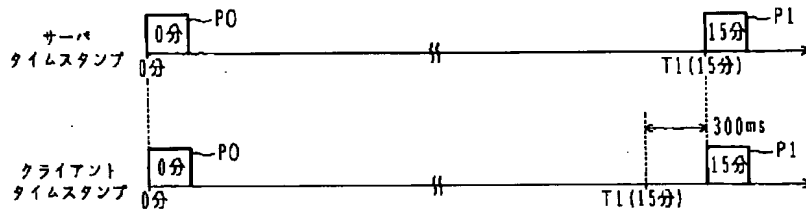
1 演奏会場、 2 MIDI楽器、 3 エンコーダー(サーバ)、 4 カメラ、 5 エンコーダー、 6 ルータ、 7 サーバ、 8 WWWサーバー、 9 ホームコンピュータ(クライアント)、 10 MIDI音源、 11 音声出力装置、 12 音声入力装置、 21 バス、 22 CPU、 23 システムクロック、 24 RAM、 25 外部記憶装置、 26 MIDI

18

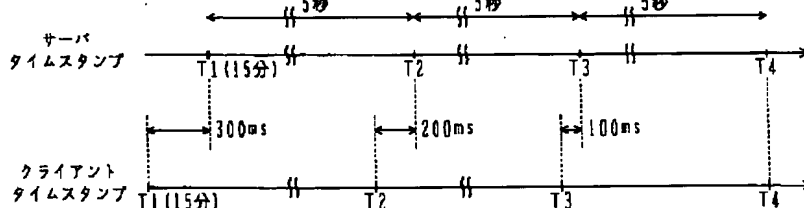
インタフェース、 27 サウンドカード、 27 a バッファ、 27 b コーデック回路、 28 ROM、 29 表示装置、 30 入力手段、 31 通信インタフェース、 32 通信ネットワーク、 33 コンピュータ、 41 タイムスタンプ、 42 識別子(ID)、 43 パケットサイズ、 44 MIDIデータ、 45、 47 MIDIイベント、 46 デルタタイム、 48 デジタルオーディオデータ、 49 MIDIデータパケット、 50 オーディオデータパケット、 51 ヘッダ、 52 フッタ、 53 シーケンスナンバ

【図1】

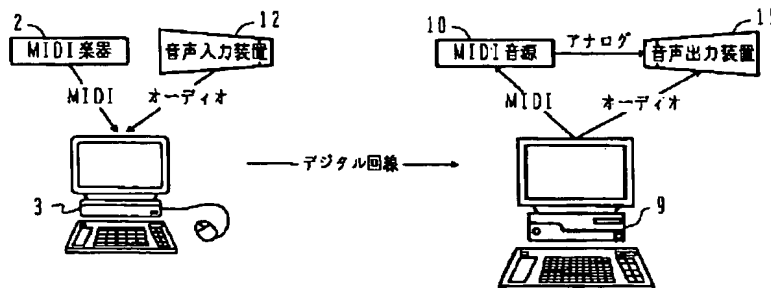
(A) タイムスタンプのずれ



(B) タイムスタンプの修正

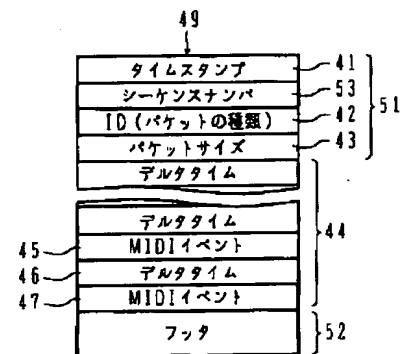


【図3】

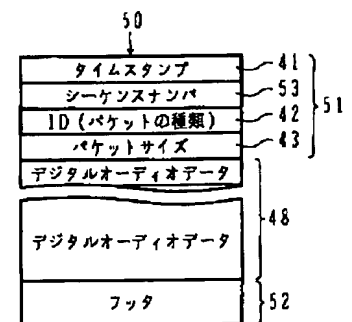


【図4】

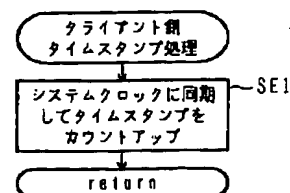
(A) MIDIデータパケット構造



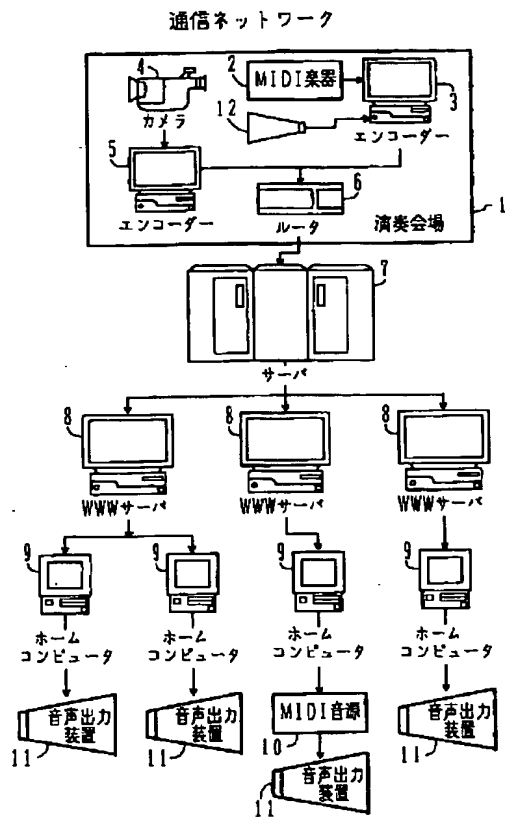
(B) オーディオデータパケット構造



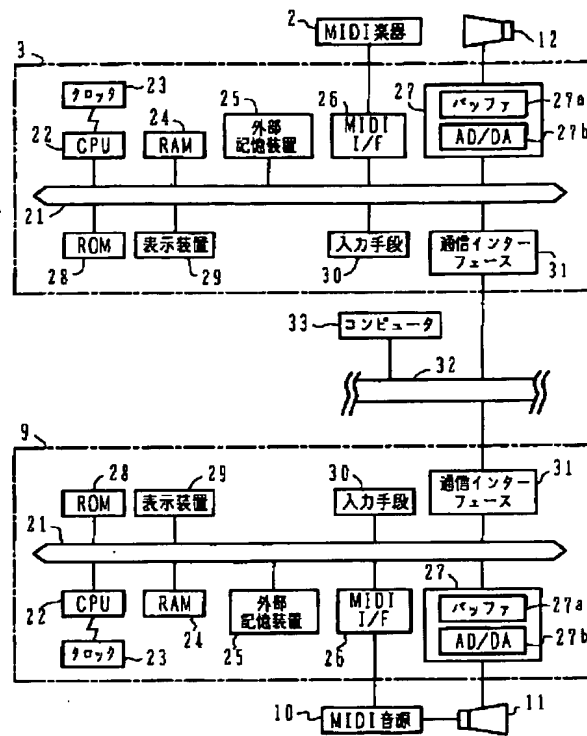
【図11】



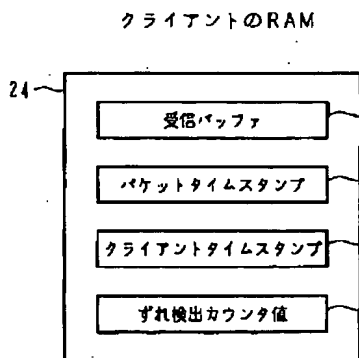
【図2】



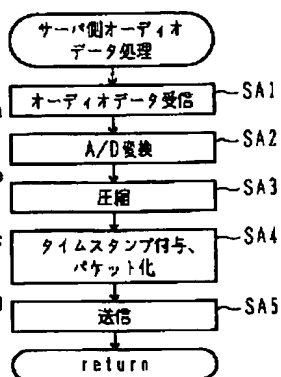
【図5】



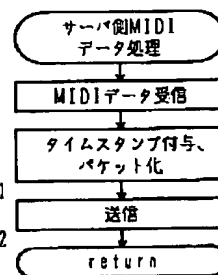
【図6】



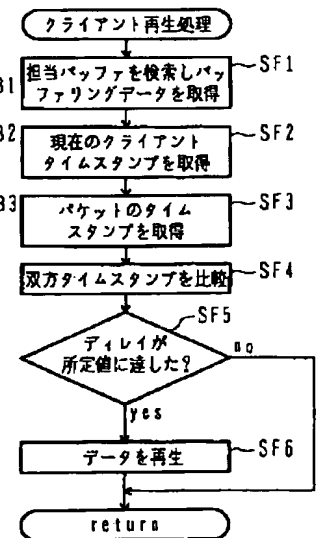
【図7】



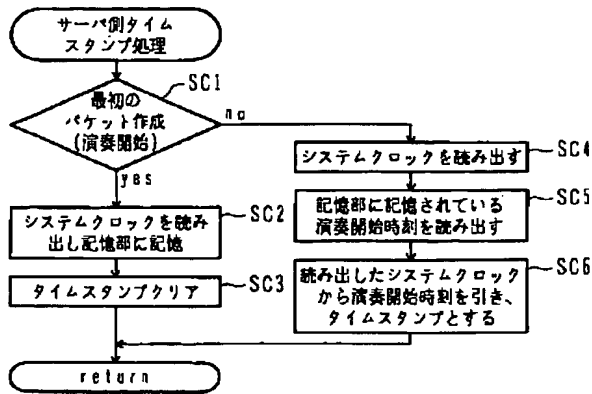
【図8】



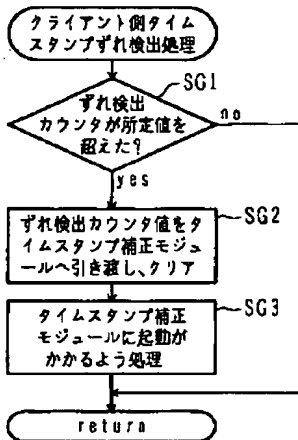
【図12】



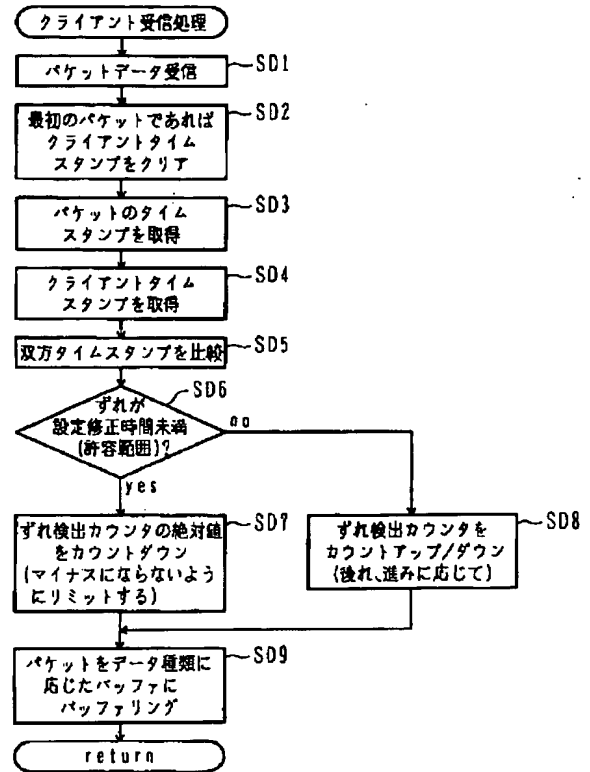
【図 9】



【図 13】



【図 10】



【図 14】

